WORK COPY - AP

(1)

- See US Pat 5901052

[19]中华人民共和国国家知识产权局

[51]Int.Cl⁷
G05B 19/408
G05B 19/042

[12]发明专利申请公开说明书

[21]申请号 98808379.5

[43]公开日 2000年9月20日

[11]公开号 CN 1267373A

[22]申请日 1998.8.19 [21]申请号 98808379.5
[30]优先权
[32]1997.8.22 [33]US [31]08/920,280
[86]国际申请 PCT/US98/17168 1998.8.19
[87]国际公布 WO99/10786 英 1999.3.4
[85]进入国家阶段日期 2000.2.22
[71]申请人 霍尼韦尔公司
地址 美国明尼苏达州
[72]发明人 杰思罗·F·斯坦曼 理查德·P·希默
M·吴拉姆·坎吉
叶海亚·C·谢哈德赫
约翰·J·罗萨－比安

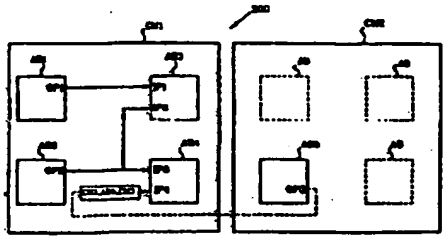
[74]专利代理机构 永新专利商标代理有限公司
代理人 韩宏

权利要求书4页 说明书22页 附图页数3页

[54]发明名称 实现分布式控制系统中算法块之间异构数据流的系统与方法

[57]摘要

本发明提供了可在控制系统中第一和第二个算法块之间实现异构数据流的一个 系统和一个方法，以及使用了这一系统和这一方法的一个分布式、实时过程控 制系统。在一个实施例中，该系统包括：(1)一个被动连接，其与一个含有 第二个算法块的一个控制模块相关联，用于提供从第一个算法块向第二个算法 块的数据传输，而不需要在第二个算法块中分配连接器资源；以及(2)一个主动连接，其与第二个算法块相关联，用于从第一个算法块向第二个算法块的 数据传输，并在第二个算法块中使用了专用的连接器资源，因而控制系统具有 既可通过被动连接也可通过和主动连接向第二个算法块提供数据的能力。



ISSN1008-4274

知识产权出版社出版

00.02.22

权 力 要 求 书

1. 一个可在某一控制系统中的第一和第二算法块之间实现异构数据流的系统，包括：

一个被动连接，它与一个含有上述的第二个算法块的一个控制模块相关联，可提供从上述的第一个算法块到上述的第二个算法块的数据传输，而无需在上述的第二个算法块中分配连接器资源；以及

一个主动连接，它与上述的第二个算法块相关联，可提供从上述的第一个算法块到上述的第二个算法块的数据传输，并在第二个算法块中使用了专用的连接器资源，因而上述的控制系统能够通过上述的被动和主动连接两种连接方式向上述的第二个算法块提供数据。

2. 权力要求 1 中所陈述的系统，其中，上述的被动连接所传输的数据不包括上述的数据的类型。

3. 权力要求 1 中所陈述的系统，其中，上述的被动连接所传输的数据不包括上述的被动连接的状态。

4. 权力要求 1 中所陈述的系统，其中，上述的主动连接所传输的数据包括上述的数据的一个类型。

5. 权力要求 1 中所陈述的系统，其中，上述的主动连接所传输的数据包括上述的主动连接的状态。

6. 权力要求 1 中所陈述的系统，其中，当上述的第一和第二个算法

块之间的上述的被动连接丢失时，上述的控制模块向上述的第二个算法块提供了一个故障保护值。

7. 权力要求 1 中所陈述的系统，其中，上述的控制模块包含上述的第一个算法块。

8. 用于在某一控制系统中的第一和第二个算法块之间实现异构数据流的方法，包括如下步骤：

在上述的第一和第二个算法块之间建立一个被动连接，用于从上述的第一个算法块向第二个算法块的数据传输，上述的被动连接不要求在上述的第二个算法块中分配连接器资源；以及

在上述的第二个算法块中分配专用的连接器资源，以提供一个主动连接，用于从上述的第一个算法块向上述的第二个算法块的数据传输，因而上述的控制系统能够通过上述的被动和主动连接两种连接方式向上述的第二个算法块提供数据。

9. 权力要求 8 中所陈述的方法，其中，上述的被动连接所传输的数据不包括该数据的类型。

10. 权力要求 8 中所陈述的方法，其中，上述的被动连接所传输的数据不包括上述的被动连接的状态。

11. 权力要求 8 中所陈述的方法，其中，上述的主动连接所传输的数据包括上述的数据的一个类型。

12. 权力要求 8 中所陈述的方法，其中，上述的主动连接所传输的数据包括上述的主动连接的状态。

13. 权力要求 8 中所陈述的方法，还包括当上述的第一和第二个算法块之间的被动连接丢失时，向上述的第二算法块提供一个故障保护值的步骤。

14. 权力要求 8 中所陈述的方法，其中，上述的第二个算法块包含于一个控制模块中，上述的控制模块执行上述的建立的步骤。

15. 权力要求 14 中所陈述的方法，其中，上述的第一个算法块含于上述的控制模块。

16. 权力要求 8 中所陈述的方法，其中，上述的分配步骤由上述的第二个算法块加以执行。

17. 一个分布式的、实时的过程控制系统，包括：

一系列传感器和可控的设备；

数据处理和存储电路，与上述的一系列传感器和可控的设备相关联，可用于执行软件指令序列，以在上述的控制系统中的第一个和第二个算法块之间实现异构的数据流，包括：

一个被动连接，与一个含有上述第二个算法块的控制模块相关联，用于提供从上述的第一个算法块向上述的第二个算法块的数据传输，而不要求在上述的第二个算法块中分配连接器资源；以及

一个主动连接，与上述的第二个算法块相关联，用于提供从上述的第一个算法块向上述的第二个算法块的数据传输，并在上述的第二个算法块中使用了专用的连接器资源，因而上述的控制系统能够通过上述的被动和主动连接两种连接方式向上述的第二个算法块提供数据。

18. 权力要求 17 中所陈述的系统，其中，上述的被动连接所传输的数据不包括上述的数据的类型。

19. 权力要求 17 中所陈述的系统，其中，上述的被动连接所传输的数据不包括上述的被动连接的状态。

20. 权力要求 17 中所陈述的系统，其中，由上述的主动连接所传输的数据包括上述的数据的一个类型。

21. 权力要求 17 中所陈述的系统，其中，由上述的主动连接所传输的数据包括上述的主动连接的状态。

22. 权力要求 17 中所陈述的系统，其中，当上述的第一和第二个算法块之间的被动连接丢失时，上述的控制模块向上述的第二个算法块提供了一个故障保护值。

23. 权力要求 17 中所陈述的系统，其中，上述的控制模块包含上述的第一个算法块。

00.02.22

说　明　书

实现分布式控制系统中算法块之间异构数据流的系统与方法

本发明的技术领域

一般情况下，本发明致力于计算与处理系统，更具体地说，主要致力于那些在一个分布式控制系统中的算法块的参数之间提供异构数据流连接的计算与处理系统以及提供实现这一连接的方法。

本发明的技术背景

自动化的工厂控制系统包括一系列可用以控制和监视工厂（例如在一家制造工厂）中各种过程的完善的算法或一系列可由软件定义的过程控制例程。可以对这一控制系统加以修改，以满足整个工厂或工厂中某些特殊部位的广泛的过程需求。传统上，控制系统包括一系列模块，每一模块都具有其自己的处理器或固件，并由传输总线将它们连接在一起，以产生一个分布式的过程控制系统。这一系统的分布特性使其具有良好的处理性能，并能够实现对系统的逐步扩展，以满足工厂设备不断增长和不断改造的需求。

自动化工厂管理的一个基本目标是提供一个能够把工厂范围的全部过程综合在一起的过程控制方案，以提高工厂的整体运行效率。一般情况下，过程控制系统可提供一种创建定制化过程策略的方法，例如可由软件定义的过程控制例程。在面向对象的程序设计环境中，一个完整的控制策略可以根据规模较小的叫做"块"、" 参数"以及"连接"的部件加以建造。一个块是一个用于封装基本控制计算的数据和算法

的软件结构;参数定义了可与块内的各数据部分相连的接口;连接将允许数据在块的参数之间流动。

一个连接的基本功能是,提供不同块的参数之间的数据流。然而,由于依赖于一个块算法的设计,仅提供简单的连续性和简单的数据流是远远不够的。不同的块算法可能要求不同的连接功能元件。例如,尽管一般情况下块算法拥有关于所传送数据的类型的隐含的信息,但某些算法可能要求提供比这些隐含的信息更多的信息,有些算法可能还需要由连接服务提供的显式的数据类型信息。

另外,某些块算法可能需实现内置的安全处理,这不仅要求关于所传送数据的值信息,而且还要求获知连接的连续性是否得以很好维持的信息。在某些情况下,区别不同的可能造成连接丢失的故障类型的状态信息也将是必须的。那些不需要显式存取状态或数据类型的块算法可能仍需要在连接性丢失时,获知所提交数据的属性。这样的块需要一个可使用的"故障保护"值,这一故障保护值对不同的数据类型可能不尽相同。

有的块算法可能要求参数,这些参数仅在少数过程控制策略或那些偶然需要连接的块中加以连接,但它们不需要显式存取数据的类型或状态。另外,某些块可能不能为每一个有时需要连接的参数分配专用的资源。全套块算法必须向用户提供一个可对参数加以连接的配置模型,那些面向实施例的算法将需要这一模型。一个用于构造过程控制方案的系统,应允许块算法之间以某种方便的方式共享数据,而且不应仅仅为了建立一个连接就要求使用附加的块。

因此,这一技术所需的是一个功能更为强大、且更为灵活的数据存取形式,这一形式应可以在一个分布式控制系统中实现算法块的参

数之间的异构的数据流连接。

发明总结

为了克服以上所讨论的事前技术中的缺陷，向控制系统中的参数传输提供一个更灵活的提供数据流的方式，是本发明的主要目的所在。

为了实现上述的主要目标，本发明提供了可在控制系统中第一和第二个算法块之间实现异构数据流的一个系统和一个方法，以及使用了这一系统和这一方法的一个分布式、实时过程控制系统。在一个实施例中，该系统包括：(1) 一个被动连接。一个被动连接与一个含有第二个算法块的一个控制模块相关联，用于从第一个算法块向第二个算法块提供数据传输，而不需要在第二个算法块中分配连接器资源或明显动作；以及(2)一个主动连接。 一个主动连接与第二个算法块相关联，用于从第一个算法块向第二个算法块的数据传输，并在第二个算法块中使用了专用的连接器资源和处理动作，因而控制系统具有既可通过被动连接也可通过主动连接向第二个算法块提供数据的能力。

当在此处所使用时，一个"连接"意味着任何可由软件加以定义的例程、或例程，以及相关的数据，用以以单个的或以组合的形式提供此处所描述的功能；一个"主动连接"通过两个所连块之一的主动服务和引用存储在该块之中的参照数据，提供了算法块之间的数据连接；一个"被动连接"提供了两个算法块之间的数据连接，而不需要任何一个所连块上的显式的动作，也不需使用存储在任何一个块中的参照数据。被动连接是通过一个控制模块的数据和服务加以实现的，控制模块包括一或两个由被动连接加以连接的算法块。

因此本发明介绍了建立多个(即"异构的")连接类型的一个更宽的概念,以反映这样一个事实:不同的算法块通常要求不同级别的参数存取。取代提供单一的、不灵活的连接类型来处理所有参数传输的做法,本发明提供了两种功能更为强大的连接方式—主动连接和被动连接。其中,允许仅当需要时和资源较为充分时才对主动连接加以建立,并允许当主动连接不需要时对被动连接加以建立。主动连接和被动连接都可由单一的一个算法块加以使用,从而控制系统可同时针对处理过程和资源的利用率加以优化。

在本发明的一个实施例中,由一个被动连接所传输的数据不包括该数据的类型,而一个算法块可能拥有关于所传输数据的类型的隐式的信息。不显式地传输数据类型可减少专门用于算法块之间数据传输的系统资源的数量。相类似,在一个实施例中,由被动连接所传输的数据不包括该被动连接的状态(例如可用性)。然而,在一个相近的实施例中,当第一和第二个算法块之间的被动连接丢失(即不可用)时,控制模块向第二个算法块提供了一个故障保护值。 那些熟悉这一技术的人将深知在控制系统中传输故障保护值的重要性。本发明甚至还允许故障保护值与被动连接间的传输。

在一个实施例中,由主动连接所传输的数据包括该数据的类型。在一个相近的实施例中,由一个主动连接所传输的数据包括该主动连接的状态。向算法块提供传输数据类型或一个连接的状态的能力,需要分配额外的系统资源,然而在使用这种算法块的控制过程的设计中提供了更大的灵活性。

在一个实施例中,包含第二个算法块的控制模块还包括第一个算法块,一个控制系统可能使用多个拥有相关算法块的控制模块。本发

明考虑到从一个算法块向第二个算法块的数据传输，使用了一个被动连接或主动连接，而不管第一个算法块是否与第二个算法块包含在相同的控制模块中。

在本发明的一个实施例中，控制系统是一个分布式的、实时的过程控制系统。然而，那些熟悉这一技术的人可以理解本发明在各种类型控制系统中的其它形式的应用。

以上相当粗略地勾画出了本发明的特性与技术上的优点，以致于那些熟悉这一技术的人可以更好地理解以下对本发明的详细的描述。更多的特性和优点（构成了本发明权力要求的主体）将描述如下。那些熟悉这一技术的人将会对本发明理解，因为他们可以很容易地把本发明所透露的概念和实施例作为改进和设计其它结构的基础，实现与本发明相同的目标。那些熟悉这一技术的人还将会体会到，就本发明的最主要的形式而言，这些等价的结构将不会背离本发明的基本宗旨。

对附图的简要描述

为了更好地理解本发明以及本发明的优点，以下将结合附图对本发明加以描述。图中同一部件具有相同的标号，其中：

图 1 描述的是一个适于使用本发明的示例性的分布式、 实时过程控制系统的功能图。

图 2 描述的是一个示例性数字处理系统的高层方框图。 这一系统能够用于执行体现了本发明基本原理的可由软件定义的过程控制例程。

图 3 是对使用了本发明的基本原理的一个示例性的控制策略应用

的图示说明。

图 4 是对图 3 中所示的示例性控制策略应用的一个连接子系统图的图示说明。

详细的描述

在对本发明的系统和方法的一个示例性的实施例进行描述之前，先对一个适合使用或实现本发明的计算或处理系统环境加以描述将是十分有益的。首先请参照图 1。图 1 所描述的是一个适于使用本发明的示例性的分布式、实时过程控制系统(在以下的整个描述过程中将其标记为 100)的功能图。

过程控制系统 100 直观地包括一个拥有一台服务器 110 和一个控制器网络 111 的计算机网络。控制器网络 111 在服务器 110 和过程控制器(在以下的整个描述过程中将其标记为 121)之间提供了一个接口。例如，控制器网络 111 在服务器 110 和过程控制器 121 之间传送控制消息，并在各过程控制器 121 之间传送对等消息。过程控制器 121 通过一个输入/输出("I/O")网络 112 与 I/O 设备 122 进行通信。使过程控制器 121 适合执行一系列可由软件定义的过程控制例程，以通过 I/O 设备 122 和 I/O 网络 112 控制和接收来自过程传感器和传动装置 130 的数据。那些熟悉这一技术的人对各种类型的可用于众多产品制造过程的传感器和传动装置 130，例如电控马达、阀门、泵等将不会感到陌生。本发明的基本原理不局限于某一具体的过程或一个具体的处理系统，而且在任何这样的系统都能够很容易对本发明的基本原理加以使用，并受益于本发明所提供的好处。

在一个实施例中，过程控制系统 100 还包括一个局域网络 113，提

供了服务器 110 和远程工作站(在以下的整个描述过程中将其标记为 140) 之间的一个接口。远程工作站 140 可由系统操作员加以使用,用以控制和监视过程控制系统 100 的操作。尽管被表示为一个独立网络,但 LAN112 和控制器网络 111 也可以是相同的,即远程工作站 140 和过程控制器 120 可以共享同一网络传输媒体。然而,那些熟悉这一技术的人都知道: 为控制系统和操作员工作站提供分离的网络可以提高一个分布式、实时过程控制系统的可靠性,例如,在 LAN 112 上的、与从服务器 110 到操作员工作站 140 的分布式相关处理数据相关联的网络交通不会干扰服务器 110 和远程过程控制器 120 之间通过控制器网络 111 所传输的过程控制信息。

可由软件定义的过程控制例程能够由任何一种数字处理系统(例如服务器 110、工作站 140 或过程控制器 121)加以执行。图 2 描述的是一个示例性数字处理系统 200 的高层方框图。这一系统能够用于执行体现了本发明基本原理的可由软件定义的过程控制例程。示例性的数字处理系统 200 包括一个微处理器 210、永久性内存 220 以及随机存取内存("RAM")230 。 能够用于存储可由软件定义的过程控制例程的永久性内存 220 可以是可编程的只读内存("PROM")、RAM 闪存或磁存储介质。存储在永久性内存 220 中的可由软件定义的过程控制例程能够由微处理器 210 加以执行。当过程控制例程得以执行时,微处理器使用 RAM230 存储所有或部分过程控制例程, 以及存储那些与过程传感器和传动装置 130 相关的过程控制数据。对示例性的数字处理系统 200 的描述仅仅是说明性的,那些熟悉这一技术的人将会体会到, 使用了本发明基本原理的可由软件定义的过程控制例程不局限于针对数字处理系统 200 的具体的硬件实现,所有这样的系统都将落入

较后所要陈述的权力要求的范畴中。

本发明提供了可在分布式控制系统中的算法块之间实现"异构"数据流的系统和方法，异构的数据流需考虑两种类型的连接服务(在此定义为"主动的"和"被动的")的可用性。主动连接通过对主动连接器的使用得以建立，它们将把由块设计者所分配的资源做为处于一个块中的对象加以处理。除了可提供基本的数据流外，主动连接器还可提供关于所引用的参数的数据类型的完整信息，以及关于所维持的连接的状态的完整信息。主动连接器之所以被叫做"主动的"，是因为在执行功能的过程中，它们要求算法块的主动的介入。另外，它们也被称为"内部的"连接，因为它们依赖于在一个块内所分配的资源。由于它们要求专用的资源，所以建议不要为由一个块所支持的每一个参数分配主动连接，而只是为一个块的大多数或所有应用中所连接的那些参数分配主动连接。与主动连接相比，被动连接允许算法块的参数在不必于一个算法块中提供专用的连接器资源的情况下，拥有数据流的连接性。

被动连接可服务于大多数参数，因为大多数参数不要求专用的连接器资源。被动连接提供了数据流，但不提供主动连接所提供的数据类型和状态服务。一个被动连接可以通过一个特殊的、叫做"控制模块"的封装块的服务得以建立。控制模块既可做为算法块的容器，也可做为连接这些块的参数的被动连接的容器。之所以把它们叫做"被动的"，是因为在执行功能的过程中它们不要求一个算法块的主动介入。被动连接也称为"外部"连接，因为连接所使用的资源是作为控制模块的一部分，在这一控制模块所包含的任何算法块的外部加以分配的。

本发明的基本原理由一系列概念和结构组成。建议最好通过可由

软件定义的过程控制例程加以实现，通过使用两个直观化的图可对这一建议更好地加以理解。第一张直观化的图(参照图 3 中的描述) 是一张展示可由某一过程控制技术人员所构造的控制策略的拓朴外观图。第二个直观化的图(参照图 4 中的描述)使用了一个"类图" 来说明一个内部软件的设计，这一内部软件可由一位系统软件设计人员加以构造。

现在转向图 3。图 3 是对使用了本发明的基本原理的一个示例性控制策略 300 的图示说明。示例性的控制策略 300 包括控制模块 CM1 和 CM2。控制模块 CM1 包含 4 个算法块，AB1、AB2、AB3 以及 AB4。控制模块 CM2 包含算法块 AB5。本发明的基本原理不局限于由算法块所实现的任何一个具体的算法，也不局限于一个具体的过程控制策略所必须的控制模块或算法块的个数。例如，控制模块 CM2 可能包含更多的算法块(在以下的整个描述过程中将其标记为 AB)。

每一算法块定义了一系列的参数，这些参数可能是输入参数或输出参数，也可能包括这两种参数。一般情况下，当创建过程控制策略时，可能会在这些块上暴露多个或仅少数几个与一个算法块相关联的参数。在控制模块 CM1 中，算法块 AB1 和算法块 AB2 分别拥有输出参数 OP1 和 OP2，算法块 AB3 拥有输入参数 IP1 数和 IP2，算法块 AB4 拥有输入参数 IP3 和 IP4。在控制模块 CM2 中，算法块 AB5 拥有输出参数 OP3。

以下，为了使参数的指示符不产生二义性，参数指示符前被冠以块指示符。当必要时，也可能向它们冠以控制模块指示符，并把句号(".") 作为参数指示符与块指示符或控制模块指示符之间的分隔符。例如，在控制模块 CM1 的环境内，可把输出参数 OP1 标记为 AB1.OP1，

把输入参数 IP3 标记为 AB4.IP3。 在控制模块 CM2 的环境内,可把输出参数 OP3 非二义性地叫做 AB5.OP3。在控制模块 CM2 的环境外, 可把输出参数 OP3 称为 CM2.AB5.OP3。

示例性的控制策略 300 把控制模块 CM1 中的连接描述为输出参数和输入参数之间的连线,数据流的方向对应于箭头所显示的方向。控制模块 CM1 中的连接是 AB1.OP1 到 AB3.IP1、AB2.OP2 到 AB3.IP2 以及 AB2.OP2 到 AB4.IP3。控制模块 CM1 还包含一个与其环境之外的一个参数的连接,即与参数 CM2.AB5.OP3 的连接。与 CM2.AB5.OP3 的连接由一个附在输入参数 IP4 上的方框加以表示,并把其称为 CM2.AB5.OP3 到 CM1.AB4.IP4。

在示例性的控制策略 300 中所描述的输入参数中,假定一个参数,AB3.IP2,要求一个主动连接,并假定实现算法块 AB3 的算法是这样的:从 AB2.OP2 向 AB3.IP2 所传输的数据类型并非总是隐式的可知的。另外,为了便于说明,还假定算法块 AB3 中所实现的算法使用了一个特殊的状态检测和响应机制,以便减轻通过使用更多算法块实现这一功能的应用设计人员的负担。参数 AB3.IP1、AB4.IP3 以及 AB4.IP4 不要求特殊的支持,于是,把这些输入参数附接于相应输出参数的连接是被动连接,它们的功能不需要算法块 AB3 和 AB4 方面的显式的动作。

当示例性的控制策略 300 所描述的过程控制例程得以执行时, 由本发明所提供的服务导致数据在连接上流动。更具体地说,控制模块 CM1 的执行导致每一算法块按照应用设计人员所指定的一个顺序依次执行,例如 AB1、AB2、AB3、以及 AB4。于是,控制模块 CM1 首先导致算法块 AB1 的执行,并运行它的处理过程,这为输出参数 OP1 产生一个

新的值，接下来又将致使算法块 AB2 的执行，并运行它的处理过程，然后产生输出参数 OP2 的一个新的值。

在算法块AB3执行之前，控制模块CM1认出连接AB1.OP1到AB3.IP1是一个被动连接，并读出 AB1.OP1 的值，然后把这一值存储于 AB3.IP1 中。接下来控制模块 CM1 致使算法块 AB3 的执行，并运行它的处理过程。算法块 AB3 认出输入参数 IP2 拥有一个相关的主动连接器，并要求特殊的处理。算法块 AB3 调用与 IP2 的主动连接器相关联的服务，读取数据的值，类型与/或状态。由算法块 AB3 所实现的算法在执行它的功能时，使用了通过一个被动连接提供的 IP1 的值，以及通过一个主动连接提供的 IP2 的数据值、类型与/或状态，即算法块 AB3 既使用了"主动"也使用了"被动"，即"异构"的数据。

在执行算法块 AB4 执行之前，控制模块 CM1 认出参数 AB4.IP3 和 AB4.IP4 是由被动连接加以服务的、于是读取 AB2.OP2 的值、将这一值存储于 AB4.IP3 中，　然后读取 CM2.AB5.OP3 的值，并将其存储于算法块 AB4.IP4 中。接下来控制模块 CM1 导致算法块 AB4 的执行，并运行它的处理过程，在这一期间，在执行它的功能的过程中，它读取并使用 IP3 和 IP4 的新值。

以上针对示例性的控制策略 300 所描述的动作，在控制模块 CM1 的每次执行时反复出现。正常情况下，连接性得以维护，且当需要时把数据提交给算法块。然而，当非正常情况出现时，连接性将会丢失，例如，可能出现使算法块 AB2 不能提供输出参数 OP2 的一个可用的值的情况。但如果这一情况发生，算法块 AB3 可以通过一个状态值发现这一情况，这一状态值是由与输入参数 IP2 相关联的主动连接器加以提供的。 在另一种情况下，与输出参数 OP3 的值的连接性(这一连接性

将来自控制模块 CM1 之外)可能丢失,此时,算法块 AB4 不具有可从其处获得状态的主动连接器。然而,在本发明的一个具体装置中,算法块 AB4 可在这样的假定下执行:当 OP3 和 IP4 之间的连接性丢失时,一个可以使用的"故障保护"值将由控制模块 CM1 提交给 AB4.IP4。

现在转向图 4。图 4 所示的是一个针对图 3 中所示示例性的控制策略应用 300 的一个连接子系统图 400。连接子系统图 400 使用了一个"类图"来说明内部的软件设计,这内部的软件可以由一位系统软件设计人员加以构造。那些熟悉这一技术的人都清楚这一类图的表示,这种表示是由 GradyBooch 在其"Object Oriented Analysisand Design With Applications"一书中加以描述的, 该书于 1994 年由 Benjamin/Cummings 出版公司出版,现将其并入此处以作参考。连接子系统图 400 按一个类图,参照图 3 中所示的示例性的控制策略 300,说明了以上所描述的行为,包括了在本发明的一个软件实现中所使用的类。 表 1 描述了这些类及它们与其它类的制约关系,并且还列出了出现在示例性的控制策略 300 中的这些类的每一取例。

表 1

类	描述	所包含的实例:	在示例性的控制策略 300 中所含的对类的实例
控制模块	实现控制模块所需的数据和算法。这些算	1 个伪代码程序 N 个算法模块 0..N 个连接器	CM1、CM2

00.02.22

	法将用于包含在控制模块中的算法块的执行和被动连接的执行。一个控制模块是算法块的父容器。没有父容器算法块将不能够存在。	1 个故障保护映象	
伪代码	一个指向某一控制模块的执行序列的二进制代码列表, 这些代码含有何时执行一个算法块或何时执行一个被动连接的信息, 命令它们按应用设计人员所指定的顺序加以执行。建议仅在有块需从连接	无	在图 400 中是非显式可见的

	那里接收输入数据之时,才对被动连接加以执行。		
算法块	对这一类的取例形成了基本的控制算法。根据这些算法,可构造更大的控制策略。它们依赖于一个父控制模块来激发它们的执行。当不再对主动连接加以使用时,提供从输出到输入参数的数据流。	1..N个参数 0..N个连接器	AB1、AB2、AB3、AB4、AB5
参数	算法块中的基础数据。参数可表示控制算法的配置数据、调整数据和过程数据。	无	OP1、OP2、OP3、IP1、IP2、IP3、IP4

	参数值的传输通过主动连接或被动连接得以进行。		
连接器	这一类的取例含有指示所连接的参数的参照信息，以及描述性数据，例如数据类型和状态。主动连接是通过定位在两个连接的算法块之一中的这一类的一个单取例加以实现的。被动连接是通过定位在含有所连算法块的控制模块中的这个类的两个取例加以实现的。如果所连的算法	1 个数据类型 1 个值参照 1 个状态	在图 400 中是非显式可见的

	块由不同的控制模块加以包含，那么建议被动连接由这个类的两个取例加以实现，且这两个取例都位于包含具有所连输入参数的算法块的控制模块中。		
故障保护映象	一张查找表。当一个被动连接的连接性受到破坏时，该查找表可使控制模块根据数据类型在表中推导出一个相应的故障保护值。这一故障保护值将传送给处在连接的输入端的参	无	在图 400 中是非显式可见的

	数。图 400 显示,有一个针对所有控制模块取例的故障保护映象。然而,在本发明的范围内,令每一控制模块拥有其自己的故障保护映象以及令故障保护映象通过用户配置加以赋值是完全可能的。		

那些显式取例连接器类的算法块(例如算法块 AB3) 在配置时间和执行时间拥有可以使用的数据类型,如同算法的处理过程所需要的。对于每一需要一个专用连接资源的参数来说,一个算法块将会对连接器类的一个拷贝进行取例,使其成为一个主动的连接器。

状态信息也可用于那些对连接器类进行取例的算法块。在配置时间可使用状态值进行有效性检查,或作为一个执行时间的检查,以验证连接操作的正确性。该值还可用于一个二进制检测,以检测连接的连续性,或在某些情况下,可用其存储所出现故障的故障类型信息。

由算法块所使用的连接器取例可由块设计人员在实现时间加以分

配,也可在实现期间的一个固定的时刻加以分配,或在一个于应用配置时间所确定的可变时刻加以分配。然而,在每一情况中,块设计人员都应该提供显式的程序语句,这些语句使用连接器服务来实现数据流、数据类型检查以及状态验证。

在实现参数的连接性方面,那些不需要主动连接的算法块,例如AB4, 不要求对连接器类进行取例,而那些仅需要被动连接的算法块,可以仅依赖控制模块类的被动连接服务。在一个实施例中,控制模块类为它所支持的每一个被动连接取例两个被动连接器,一个被动连接用作对输出参数的参照,另一个被动连接器用作对输入参数的参照。一个被动连接器定义了应把哪一个输出参数传送给哪一输入参数,而伪代码相对每一算法块的执行情况定义了传输所要进行的时间。

建议由一个控制模块类所分配的被动连接器和伪代码最好为可变的,即算法块或控制模块的设计不应限制在配置时间可与具体算法块的参数连接的被动连接的数目。由于被动连接服务是由控制模块类加以提供的,所以应用设计人员能够虚拟地连接任何算法块的任何参数,而不管块的设计者是否事先考虑到对一个主动连接器的支持。

作为其执行被动连接任务的一部分, 一个控制模块的取例可监视连接的连续性。在出现故障时,控制模块将根据它的被动连接器数据,在故障保护映象类中查找相应的故障保护值,来识别这一连接上的数据流的类型,并把这一故障保护值传送给输入参数。这一设计确保了对那些不使用主动连接的算法块的最起码的安全。故障保护映象类把故障保护值赋给具体的数据类型的方法可能随针对本发明所选择的实现方法的不同而各异:对于符合 IEEE 浮点标准的浮点数据类型来说,可以使用自然的故障保护值 NAN(不是一个数字)。 对于布尔数据类

型来说，"False"将是一个可接受的故障保护值。另外，故障保护值也可由系统用户加以配置。那些熟悉这一技术的人将会很容易地寻找到其它合适的标准，为一个具体的应用选择故障适当的故障保护值。

为了进一步说明本发明，表 2 和表 3 分别提供了两段用高度抽象的、类似 C 的程序设计语言("PDL")编写的程序。表 2 中的 PDL 对应于控制模块类的一个示例性的执行算法，表 3 中的 PDL 对应于一个算法块类的执行算法。

表 2 中的 PDL 假设：包含在一个控制模块的连接器的取例中的参照数据已作为控制模块建立操作的一部分进行过预处理。在预处理工作完成之后，伪代码已从一个一般的形式转换成一个具有完全限制的内存参照的特殊的形式。

表 2

```
PpseudoCodeInstruction=&FirstPseudoCodeInstruction;
While(pPseudoCodeInstruction<=&LastPseudoCodeInstruction){
  Switch(*pPseudoCodeInstruction){
  Case ExecuteInstruction:
     调用部件算法块的执行函数功能;
     Break;
  Case TransferInstruction;
     从*pPseudoCodeInstruction 读出参数的源地址;
     if(所读的参数招致一个错误){
        用故障保护值替换源值
}
   从*pseudoCodeInstruction 读出参数的目标地址;
```

```
        把源值传送于目标地址;
        break;
    default://EndInstruction
        从控制模块的执行功能退出;
    }
}
```

一个过程控制策略的实现通常要求多个算法块类,每一个都具有其自己的执行算法。表 3 中的 PDL 假设一个算法块,例如 AB3, 拥有一个未使用主动连接器的输入参数 IP1 和一个使用主动连接器的输入参数 IP2。针对输入参数 IP2 的主动连接器叫做"IP2Connector",且假设连接器的取例 IP2Connector 已作为算法块建立操作的一部分进行过预处理。在预处理之后,IP2Connector 拥有一个完全限制的内存参照以及数据类型和状态信息。

表 3

```
IP2Stautus=从 IP2Connector 读到的状态值;
if((IP2Status 是坏的)或(IP1 为故障保护值)){
执行相应的故障处理;
}
else{
IP2DataType=从 IP2Connector 读到的数据类型;
POutputParameter=
从针对 IP2 的主动连接器读到的参数的地址值;
```

```
IP2=
从 pOutputParameter 读到的值,并把数据类型存入 DataType;
使用 IP2 和 IP1 的值执行算法的计算;
}
```

从以上的描述中,那些熟悉这一技术的人将会认识到,无论是主动连接还是被动连接都可在一个过程控制系统加以提供,从而异构的数据流可以在算法块之间加以提供。拥有这两种类型的连接,将允许算法块的设计者控制资源的分配,与此同时还可从块参数连接中获得全部所需的功能。如果一个设计中的块需要显式的数据类型和状态,则可通过在这一块中分配一个专用的资源以使数据类型和状态能够得以使用。相反,如果就可用的系统资源而言,这一专用的资源过于昂贵,或如果所有的参数连接性事件不可预测,那么仍可使用被动形式的连接。

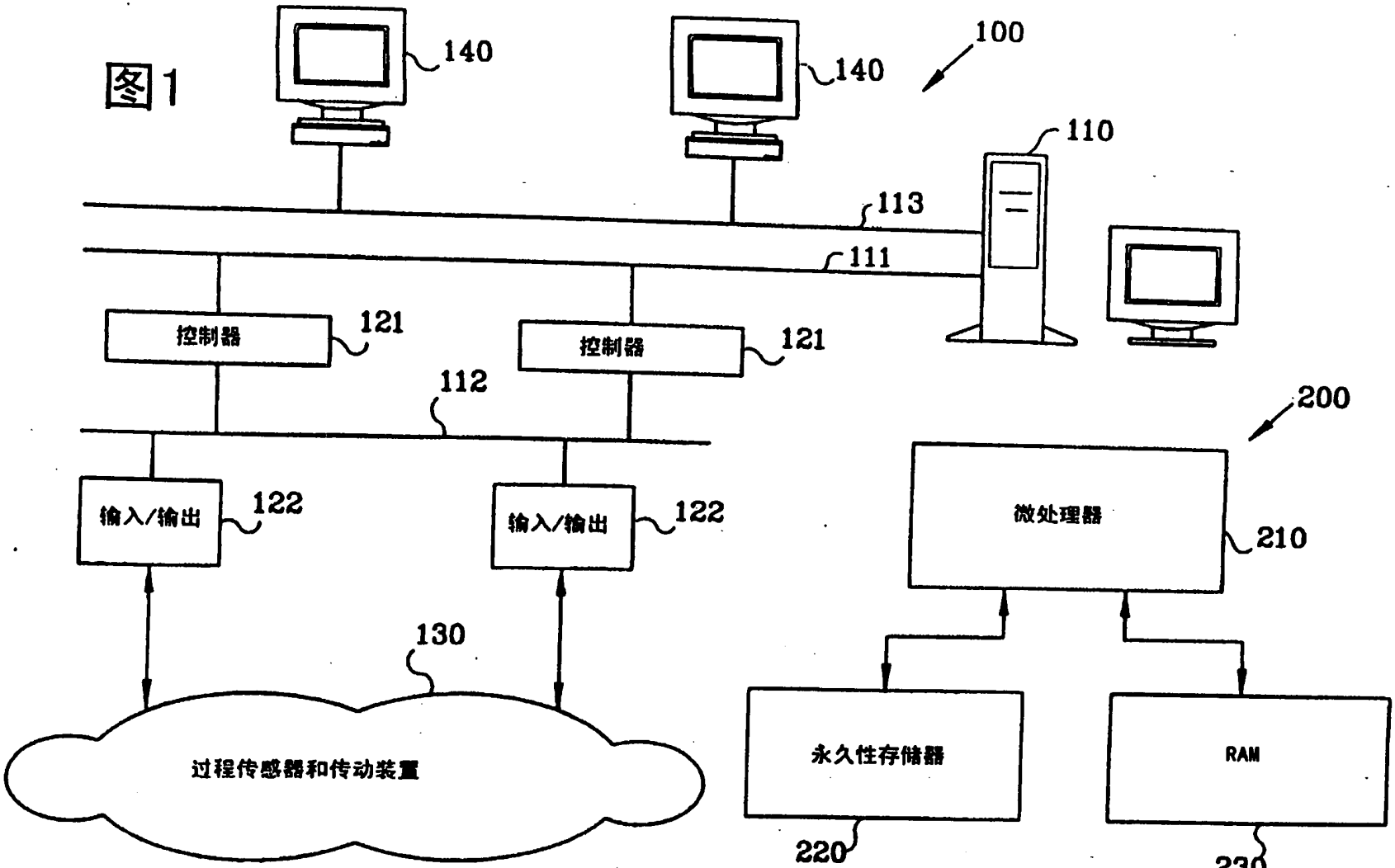
由本发明所提供的主动连接在某些方面优于事前技术,例如,它们可在块的执行期间使数据类型和状态能够得以使用。而且,若在一个面向对象的框架中加以实现,它们还可大大减轻块设计者为实现主动连接的程序设计的负担。另外,如果在算法的实现中数据类型属性是有用的,则主动连接不仅允许在执行时间,而且也允许在配置时间对数据类型属性加以存取。

由本发明所提供的被动连接在某些方面也优于事前技术。例如,在所连接的输出和输入参数之间提供基本数据流的同时,被动连接也能够提供对数据类型来说至关重要的、可定义的故障保护行为。另外,还提供了对布尔、整数以及浮点数据类型之间的被动连接的支持。对

于每一数据类型，都将会获得一个相应的可定义的故障保护值。连接性的丢失将导致故障保护值向所连输入参数的提交。

从以上的描述可以明显地看出，本发明提供了可用于在控制系统中第一和第二个算法块之间实现异构数据流的一个系统和一个方法，以及使用了这一系统或这一方法的一个分布式的、实时的过程控制系统。尽管本发明及其优点已详细加以描述，但那些熟悉这一技术的人都清楚，他们可在不背离本发明基本宗旨的情况下，对本发明的实施例进行各种变通，替换以及改动。

图1


140
140
100
110
113
111
控制器
121
控制器
121
112
输入/输出
122
输入/输出
122
130
过程传感器和传动装置

200
微处理器
210
永久性存储器
220
RAM
230

图2

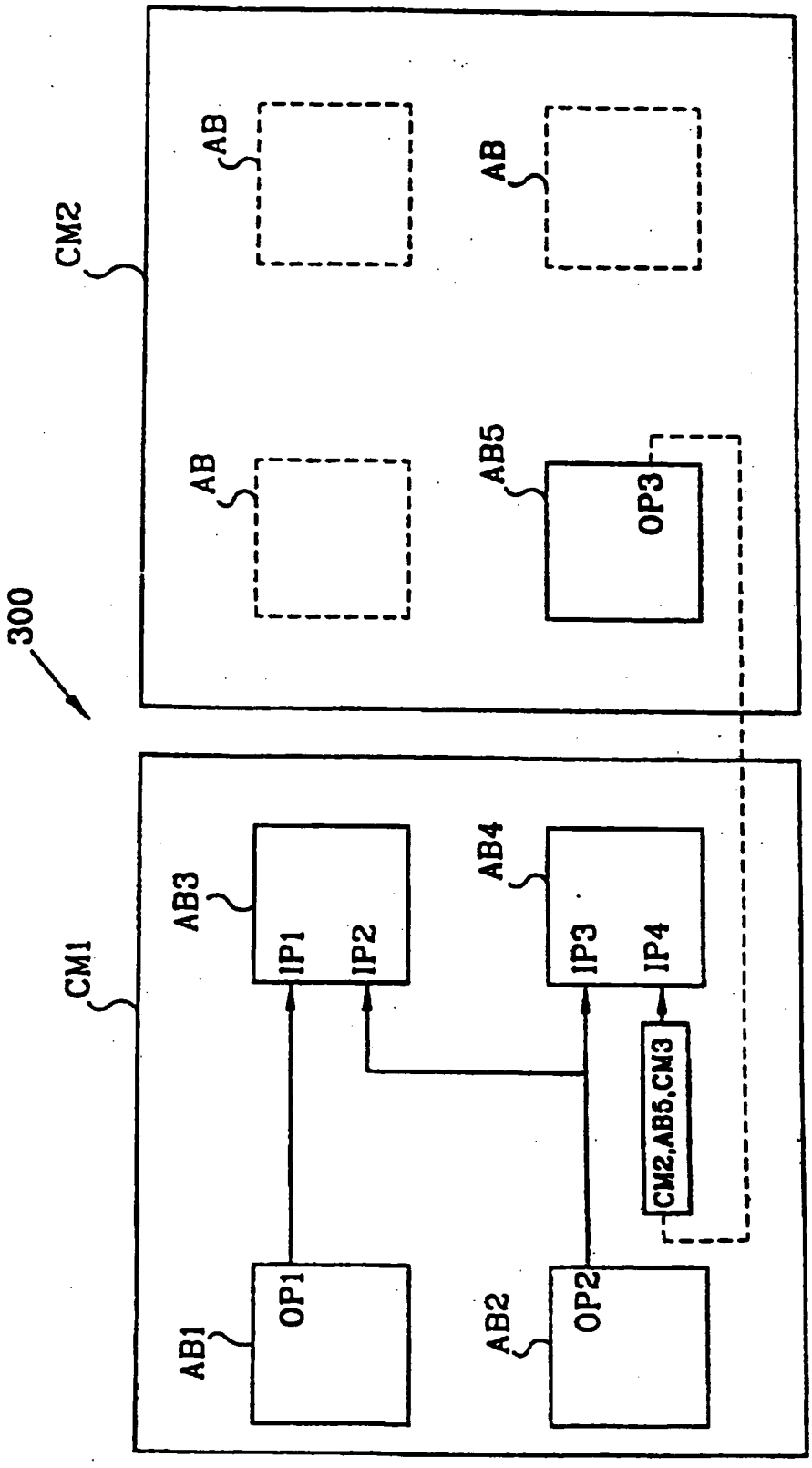

300
CM2
AB
AB
AB
AB5
OP3
CM1
AB3
IP1
IP2
AB4
IP3
IP4
CM2,AB5,CM3
AB1
OP1
AB2
OP2

图3

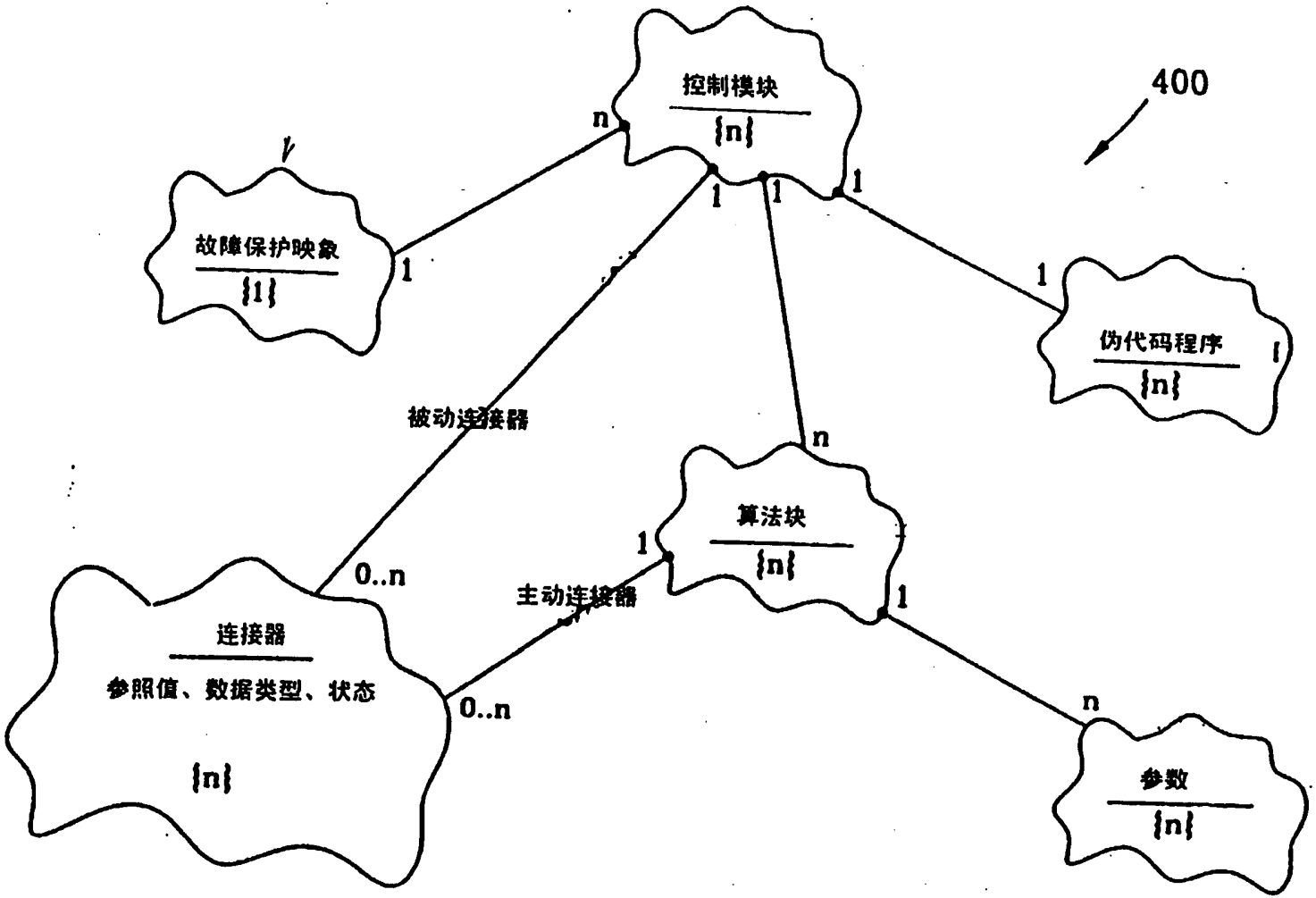

400
控制模块
{n}
故障保护映象
{1}
伪代码程序
{n}
被动连接器
算法块
{n}
主动连接器
连接器
参照值、数据类型、状态
{n}
参数
{n}
n
1
0..n

图4

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☑ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.